



INRIA49.FRD

77440.616

Apparatus for simulating the deformation of materials,
notably of soft body tissues

The invention relates to the simulation of deformation of materials, notably of soft body tissues.

Usually, a display interface is provided for representing at least one object following a predetermined shape, displayed with a selected grid pattern. A user interface provided with mechanical handling means allows the user to simulate one or more forces exerted globally on the object.

A memory zone is also provided for storing :
mechanical parameters of the material of which the object
in question consists,
data as to the position of the object, recorded at the
vertices of a selected mesh, and
force data which represent, in intensity and position,
stress to be exerted on the object.

A computer capable of co-operating with the memory zone evaluates new positions of the vertices, as a function of the stress exerted and the mechanical parameters of the material. In order to display the deformation and/or the return to a resting position of the moving object, the computer calculates the evolution of the positions of the vertices over time as a function of this stress. When it is used to simulate the deformation (or cutting) of body tissues, a qualified person (such as a surgeon) is able to follow the simulation of a deformation and/or a return to a resting position.

However, simulated deformations with large shifts in position are not normally reliable.

The present invention sets out to improve the situation.

According to a first important feature of the invention, the computer comprises:

- a module capable of repeatedly determining, for each mesh element, the deviation between the current length of a side of the mesh and its resting length, and of storing the respective force data for each vertex of the mesh, and
- a module capable of repeatedly determining, for each vertex, new positional data relating to this vertex as a function of the composition of the forces exerted thereon, data relating to at least one earlier position of the vertex and mechanical parameters of the material.

In a preferred embodiment, the mesh pattern chosen comprises at least one triangular base.

Thus, if the object is in the form of a hollow three-dimensional envelope, the mesh pattern chosen is preferably triangular, so that the computer is capable of determining the composition of the forces at each vertex of a triangle, as a function of the deviation between the current length of each side of the triangle and the length of this side at rest.

If the object has a solid three-dimensional shape, the mesh pattern chosen is preferably tetrahedral, so that the computer can determine the composition of the forces at each vertex of the tetrahedron, as a function of the deviation between the current length of each edge of the tetrahedron and the length of this edge at rest.

Advantageously, the memory zone is arranged so as to store, in conjunction with each mesh element, mechanical parameters of the material of the mesh, at least partly defined locally, particularly at the level of the mesh or elements thereof.

In the published application no. FR-9714506 by this Applicant, the simulation of the deformations of the object was based on a linear model, in the embodiment described. Although the use of the apparatus described in FR-9714506 has proved very promising in the field of surgical assistance for simulating the cutting of soft body tissues (liver, skin, etc), the simulation of large movements, possibly non-linear, was not completely natural. Moreover, this apparatus could not predict the reaction to a force exerted on just any type of deformable material, particularly in the case of large movements.

In a preferred embodiment of the present invention, the computer is designed to determine said deviation between current length and resting length, with a view to estimating a derivative of the potential energy of deformation of each mesh, expressed as a function of a Green-St Venant tensor and mechanical coefficients inherent in the material in this mesh.

The derivative of the potential energy thus expressed in relation to the position of the vertex makes it possible to obtain the force (or composition of forces) exerted thereon.

In estimating the derivatives of the potential energy at the vertices, thus expressed as a function of a Green-St Venant tensor, the computer is adapted to determine more precisely the differences between the squares of the

current lengths and resting lengths of each side, with a view to determining the composition of the forces mentioned above.

The mechanical parameters mentioned above advantageously comprise the Lamé coefficients of the material in each mesh in question.

The apparatus according to the invention can thus be used to simulate large movements and/or non-linear elasticity, possibly of heterogeneous objects, formed of different materials.

It is thus possible to simulate the deformation of a solid three-dimensional object, possibly enveloped in a hollow skin. In this case it will be advisable to apply a tetrahedral grid to the object and a triangular grid to the skin and to apply suitable mechanical parameters to each type of mesh. It is also possible to simulate the movements of soft objects (such as muscles) in the presence of rigid integrated objects (such as bones), as will be seen hereinafter.

The present invention may also take the form of a process for simulating the deformation of objects, particularly of soft body tissues, comprising all or some of the following steps :

- a) specifying a predetermined form for the object at rest,
- b) making a grid image of said form, based on a selected pattern, preferably with a triangular base (triangle or tetrahedron),
- c) storing the positional data for the vertices of the mesh in the object at rest,
- d) simulating one or more forces exerted globally on the object (or the movements of the object),

e) estimating, as a function of the mechanical properties of the material and the positions of the preceding vertices, a resultant force on each vertex of each mesh, and

f) as a function of the force exerted on each vertex, determining the movement and, hence, the deformation and/or the return to a position of equilibrium of the vertices of the meshes.

Step c) also involves attributing indices to the elements of the meshes, within the scope of establishing a data structure, and, if appropriate, simplifying the structure of the object, as will be seen hereinafter. Steps e) and f) are based on a model developed by the Applicant which will be described in detail hereinafter.

Further features and advantages of the invention will become apparent from studying the detailed description that follows and the accompanying drawings, wherein:

- figure 1 diagrammatically shows an apparatus according to one embodiment of the invention,
- figures 2A and 2B diagrammatically show triangular meshes, in the initial and deformed state, respectively, with notations for the points of the triangle subsequently used in estimating the composition of the forces at each vertex of the triangle,
- figure 3A is a diagram of a tetrahedron which helps in the understanding of the notations used for a point P_j of the tetrahedron and the normal vector \mathbf{n}_j of the opposite face,
- figures 3B to 3D are diagrams of tetrahedrons which,

together with the abovementioned figure 3A, help to show how the sums on the edges and pairs of edges of a tetrahedron are calculated, in order to evaluate a force to which a vertex P_p of the tetrahedron is subjected,

- figures 4A and 4B are diagrams to assist in the understanding of the attribution of indices in different triangular meshes,

- figures 5A and 5B are diagrams of triangular motifs comprising, respectively, a current vertex (represented by a square) and a virtual vertex (represented by a disc), before simplification, and the same gridded shape after simplification,

- figure 6 diagrammatically shows an example of a non-conforming triangular mesh, and

- figure 7 shows a diagrammatic flow chart of the progress of the calculations of forces and positions of the vertices representing the object.

The present description will refer to formulae and equations which appear in the annex.

The drawings, and the annex, essentially contain elements of a definite nature. They can therefore not only be used to gain a fuller understanding of the description, but may also contribute to the definition of the invention, where appropriate.

Reference is made first of all to Figure 1 in which the apparatus is shown in the embodiment described in the form of a computer (PC). The computer comprises a permanent memory MEM (for example of the ROM type) in which are

stored programmes for calculating the positions of the mesh vertices and the resultant forces at these vertices, as well as a processor μP capable of co-operating with a working memory MT (for example of the RAM type) for carrying out these calculations.

The abovementioned modules of the computer described hereinbefore may, if necessary, be in the form of programmes stored in the memory MEM, or may be loaded into the working memory MT, from reading a data carrier in an external peripheral (not shown) such as a disc drive or a CD-ROM reader, or received via a communications link, e.g. a MODEM. Of course, such programmes appear to be particularly important means for carrying out the invention. Thus, the present invention also relates to programmes of this kind, possibly on a data carrier of the type mentioned above (disc, CD-ROM, etc), detailed embodiments of which are described hereinafter.

The computer further comprises a monitor VISU connected to the processor μP via a graphic interface IG, the unit consisting of VISU and IG (referred to as the "display interface" hereinbefore) providing a display of an object the deformation of which is being simulated. The computer further comprises a user interface IU connected to mechanical handling means such as a keyboard CLA or a mouse MO, enabling a user to simulate a force exerted globally on the object, or a change of position of this object, with initial deformation.

The shape of the object to be displayed in grid form may be a surface shown in three dimensions. In this case, the mesh is triangular (figure 2A). The shape of the object to be displayed in grid form may be a three-dimensional solid and in this case the mesh is tetrahedral (figure 2B). If

appropriate, a grid module is stored in the memory MEM and can be implemented by the processor μP , co-operating with the graphic interface IG, to display a shape of the grid-covered object on the monitor VISU.

The following is a description of a model developed by the Applicant and applied to an apparatus for simulating deformation. This model uses a resting form without mathematical induction in relation to a resting position (mathematical induction by shifts which is normally the basis of conventional models).

Let us consider first of all a surface shown in three dimensions, gridded with a triangular mesh pattern. A triangle $T_0 = (P_0, P_1, P_2)$ of the resting form, with edges (l_0, l_1, l_2) . This triangle (on the left in figure 2A) undergoes a transformation Φ to give the triangle $T = (Q_0, Q_1, Q_2)$ (on the right in figure 2B).

The potential elastic energy of this deformation is expressed here from the Cauchy-Green C deformation tensor (formula F1 of the Annex). Looking at the finished element obtained by associating a set of linear interpolation functions with this triangle, the deformation gradient is expressed at every point X on the triangle by the formula F2, wherein A is the area of the triangle T_0 , n_j is the unit vector perpendicular to the segment (P_{j+1}, P_{j+2}) , and the α_j 's are shape vectors of the triangle.

The deformation tensor C is then given by the expression of formula F3. Advantageously, this deformation tensor is invariant by translation and therefore does not depend on the choice of the origin of the reference point. Preferably, the origin of the reference point is placed in

the centre of the circle circumscribed around the deformed triangle T, of radius r (figure 2A), which allows the tensor C to be expressed in the form given in formula F4, where L_p is the length of the deformed edge joining the vertices Q_j and Q_k .

The expression of the potential deformation energy (given by formula F5 in the annex) uses those of the trace of the tensor C (given by formula F6) and of the trace of the tensor C^2 (which will be described hereinafter).

Starting from the classic equations in the triangle T_0 (cosine rule), which are expressed by the formula F7, the scalar products between the directing vectors are obtained, given by equation R1 in the annex. The respective traces of the tensors C and E are thus written according to formulae 8 and 9 in the annex. This term depends only on the variations in length of the edges of the triangle and its resting shape (represented by the magnitudes K_i).

From the expression of $\text{tr } C^2$ (formula F10), $\text{tr } E^2$ is deduced and, finally, the potential elastic energy of deformation of the triangle is given by equation R2 in the annex.

The elastic force applied by the triangle at one of its vertices is obtained by deriving the elastic energy relative to the position of this vertex. In the embodiment shown in Figure 2A, the vertex P_0 has Q_0 for its deformed position. Deriving the equation R2 in relation to Q_0 , the first step is preferably to derive the terms L^2_p in order to obtain the expression of the force according to formula F11, which is the equation of the elastic force exerted by the triangle at each of its vertices. This force is expressed solely as a function of the variations in length of the sides of the triangle ($L_i^2 - l_i^2$), the mechanical

parameters of the material (λ and μ) and the shape of the triangle at rest (given by the terms K_i).

Advantageously, this force implicitly contains a property of rotational invariance as all its terms remain zero as long as the lengths of the sides of the triangle do not change, i.e. as long as the triangle is not deformed. Thus, the calculations of movements can be simplified by virtue of the fact that it is no longer necessary to compensate for rigid transformations, as invariance by rigid transformation is intrinsic to the formulation of the deformable model.

Extending the foregoing to a solid three-dimensional shape (having a volume), gridded with a mesh of tetrahedrons, by formulating the elasticity according to Saint Venant-Kirchhoff it is possible to express the gradient of deformation (formula F12), as well as the Cauchy-Green deformation tensor (formula F13), where A_j is the area of the face opposite the vertex P_j , n_j is the perpendicular to this face and V is the volume of the tetrahedron (Figure 3A).

By placing the centre of the reference point at the centre of the sphere circumscribed around the tetrahedron T , it is possible to express the scalar product of the position vectors of two vertices of the tetrahedron as a function of the length of the edge that joins them and the radius r of the sphere, thus making it possible to write the deformation tensor according to formula F14.

The elastic deformation energy is expressed as a function of the Green-Saint Venant deformation tensor (bearing the reference E), but in a 3D grid (formula F15). The

expression of the trace of the tensor C is also expressed with the scalar products $(\alpha_j.\alpha_k)$, using the cosine rule for dihedral angles and as a function of the lengths of the edges of the tetrahedron (formulae 16 and 17).

Consequently, the trace of C^2 is written according to formula F18. Finally, the Applicant has shown that the elastic deformation energy is expressed by the formula F19 of the annex (complemented by the formula F20).

The expression of the elastic force exerted by the tetrahedron on one of its vertices is obtained by deriving the elastic energy relative to the position of this vertex. A formulation which is easy to use is expressed by formula F21 of the annexe and is interpreted using Figures 3A to 3D.

In formula F21 :

- in the second line, k and m are the complementary indices of j and p ,
- in the last line, m is the complementary index of j , k and p , and : $K_{jkm} = K_{jp}K_{km} + K_{jm}K_{kp} - K_{jk}K_{mp}$

The total on the edges (j,p) indicates that it is necessary to travel over the three edges adjacent to the vertex P_p (on which the force is exerted) whereas the total on the edges (j,k) (indices different from p) involves the three edges located on the face opposite the vertex P_p .

Referring to Figures 3A to 3D :

- the total on the edges (j,p) involves three terms (Figure 3A),
- the total on the pairs of edges $((j,p),(k,p))$ involves three terms (Figure 3B),
- the total on the pairs of edges $((j,p),(j,k))$ involves six terms (Figure 3C) and

- the total on the pairs of edges $((j,k),(m,p))$ involves three terms (Figure 3D).

Once again, there is rotational invariance of the above expressions, for a solid three-dimensional form, which advantageously makes it possible to avoid the appearance of unrealistic distortions when part of the object is rotated too far. This invariance also extends to movements of translation. Thus, the object may undergo any overall movement of translation and rotation while retaining zero internal energy and stresses. The Applicant has verified that in this case the rigid component of the deformation is zero.

Moreover, the discretisation of the object using finished elements comes to approximate the elastic energy over a set of sub-domains defined by simple geometric shapes. The above model thus suggests, for each sub-domain, a formulation of the energy which only depends on the change in shape of this domain.

Advantageously, displaying a simulated deformation while considering an initial resting shape (rather than initial resting positions) appears to be more natural than the deformations simulated with known models. Moreover, considering an initial resting shape avoids the problems of accuracy that arise when the model moves substantially away from its resting position.

The following is a description of the preliminaries to using the abovementioned computer modules to determine the forces at each vertex of the chosen mesh.

First of all, a data structure should be defined. The Applicant has first dealt with the difficulty of

attributing suitable indices to the elements of each mesh.

For triangular patterns, the data structure corresponding to triangulation consists of a list of triangles, a list of sides and a list of vertices:

- for each vertex, an adjacent side is determined,
- for each side, a pointer is determined on the two adjacent triangles, and the two vertices which form the side are determined,
- for each triangle, the pointers are determined on the three vertices that form the triangle, the three sides of the triangle are determined, the three triangles adjacent to each side are determined, and three values giving the index of adjacency of the current triangle for each adjacent triangle are determined.

The orientation of the three vertices in the triangle is carried out as follows.

For the grid as a whole to be correctly oriented, each edge or side must be travelled over in two different directions when each of its two adjacent triangles is travelled over. Referring to Figure 4A, the side A_i of index i is the side opposite the vertex S_i of index i in this triangle. The adjacent triangle to index i is the neighbouring triangle TV , adjacent to the side A_i . Finally, each triangle always has three adjacent triangles. In order to represent surfaces with borders (a plane has one border, a cylinder has two borders) the notion of so-called "virtual" vertices and triangles is used (Figure 4B). The virtual triangles are not displayed and the position of the virtual vertex is of no consequence to the geometry of the grid.

An example of index attribution will now be given by way of example, in C++ language.

If the adjacent triangle number i is called `neighbor[i]`, then the adjacency index "`whichNeighbor[i]`" is defined by an instruction such as :

`Neighbor[i]->neighbor[whichNeighbor[i]]=this ;`

For reasons connected with a tetrahedral grid, the data structure of a tetrahedral volume consists of a list of tetrahedrons, a list of triangles, a list of edges and a list of vertices.

For each vertex, its position and an adjacent tetrahedron are stored.

For each edge, the pointers on the two adjacent vertices and a pointer on a triangle adjacent to this edge are stored.

For each triangle, the pointers on the three vertices that form the triangle and the pointers on the two tetrahedrons adjacent to the triangle are stored.

For each tetrahedron:

- the pointers on the four vertices that form the tetrahedron,
 - the pointers on the six edges of the tetrahedron,
 - the pointers on the four triangles of the tetrahedron, and
 - the pointers on the four adjacent tetrahedrons opposite each of the four vertices
- are stored.

Each tetrahedron is oriented. Starting from the orientation of the four vertices ABCD , the operator seeks to orient the four triangles : BCD, ADC, ABD, ACB. To ensure correct

orientation of the grid, each triangle has to be travelled over in two different directions by travelling over each of its two adjacent tetrahedrons. Finally, each tetrahedron is always adjacent to four tetrahedrons. The notion of so-called "virtual" vertices and "virtual tetrahedrons" is introduced to take account of the borders of the grid.

In order to simplify the structure of the deformed object, the grid is cut up by removing material (more precisely a vertex, although this vertex cannot be a virtual vertex).

In the case of a triangular mesh, if this vertex is not adjacent to a virtual vertex, this vertex is regarded as being virtual like all these adjacent triangles. If the vertex is already adjacent to a virtual vertex, the side which connects this point to the virtual vertex is removed, and all the triangles adjacent to the virtual vertex are forced to be virtual (Figures 5A and 5B).

With a tetrahedral grid pattern, the procedure involves removing a set of associated tetrahedrons. Once the tetrahedrons have been removed the grid must remain true, as described hereinafter. For this, it may be advantageous to remove more tetrahedrons than those present in the set of tetrahedrons mentioned at the beginning. Virtual vertices may also be added or removed when creating associated components or removing the borders.

The Applicant has confronted the problem of ensuring conformity of the grid, a definition of which is provided hereinafter.

For a triangular mesh to be true, each side must belong to more than two triangles and the topological neighbourhood of each vertex must remain homeomorphous with a circle (or

with a semicircle), i.e. if two triangles share the same vertex, there must be a set of triangles adjacent to each other which make it possible to go from one triangle to another. Referring to Figure 6, the triangular mesh shown is not true, as the two triangles have an intersection along a vertex, while there are no adjacent triangles making it possible to go from one of these two triangles to the other.

For a tetrahedral grid to be true, each triangle must belong to more than two tetrahedrons and the topological neighbourhood of each vertex must be homeomorphous with a sphere (or a hemisphere if it is on the border). Therefore, if two tetrahedrons share the same vertex, it is essential that there is a set of tetrahedrons adjacent to one another which make it possible to pass from one tetrahedron to another.

The equation that governs the movement of the linear elastic model is the Newtonian differential equation E1 in the annex. This equation is connected with the differential equations encountered in the mechanics of continuous media (equation E2).

According to the theory of the finished elements, the mass matrix M and the attenuation matrix C are hollow and depend on the properties of each tetrahedron. Moreover, these matrices M and C may be considered here as being diagonal, i.e. the mass and the effects of attenuation are concentrated on the vertices of the mesh. This simplification, known as *mass-lumping*, uncouples the movements of each of the nodes and thus makes it possible to write the previous equation E2 as a set of independent differential equations of the type E1, with an equation for

each vertex.

The choice of an explicit integration diagram leads to the expression of the elastic forces at the time t , to calculate the new position of the vertex at the time $t+1$, with the equation E3 in the annex. Advantageously, this equation E3 applies to all types of force, both linear and otherwise.

The explicit diagrams are only conditionally stable and therefore converge less rapidly than the implicit diagrams.

Another difficulty arises : for the diagram to be stable, the time step used for the digital integration should be sufficiently short. The critical time step depends on the greatest intrinsic value of the rigidity matrix and of the local values of the mass and of the attenuation. To optimise the time step, the mass matrix is fixed to be equal to identity, and the optimisation of the local values of attenuation, so as to limit the risks of oscillations, have the effect of improving the packaging of the linear system, and hence of stabilising its resolution.

Reference will now be made to Figure 7, wherein a flow chart diagrammatically shows the different treatment steps applied to a triangulated three-dimensional surface, by way of example.

The treatment begins with a first step 10, in which the structure of the data described above is applied, with a suitable attribution of indices. Then in step 12 consideration is given to each triangle T_n , for which the Lamé coefficients λ_n and μ_n are sought, as well as the area of the triangle A (step 14). For each side of the triangle

(j being an integer from 0 to 2), the difference between the squares of the lengths of the sides is calculated, between the current length l_j and the previous length L_j of the side (step 16). From these differences in the lengths of the sides squared, the forces exerted at each vertex S_j of the triangle T_n are deduced (force F_j^n calculated in step 18 of the flow chart). Moving on to the next triangles, the forces are integrated over all the triangles by looping on the index n (step 20).

In steps 30 and 32, the new positions Q'_n of the vertices S_j of the deformed triangles are determined, from their previous position Q_n and the forces F^n . By looping on to the index n in step 36, integration is carried out over all the deformed objects. The new positions of the vertices S_j are preferably estimated by using the equations E1, E2 and E3 mentioned hereinbefore.

In the flow chart in Figure 7, the two respective integrations (of the forces F_j^n and of the positions of the vertices S_j in the deformed objects) are shown separately. Of course, it may be that some of the steps 12 to 20 may be carried out at the same time as or after some of the steps 30 to 36. This choice depends on the structure of the two modules described hereinbefore, contained in the abovementioned computer, and, if applicable, on the data structure.

A test 42 is preferably provided to find out whether the object has fully returned to its resting position, i.e.: is there an edge in the object the length of which differs substantially from the previous one?

If the computer does not find such an edge as it travels over the triangles, the object has returned to its initial

resting shape and the treatment is stopped at 40.

Otherwise, an image of the object is displayed by the monitor VISU mentioned above and a time loop (step 44) is incremented, to pass to the next time and show a progressive return to a resting shape of the object, using equation E3, if appropriate. This time loop can be initiated after the first starting step 10.

Of course, the invention is not limited to the embodiment described previously by way of example, but also extends to other alternative embodiments defined within the scope of the claims that follow.

Thus, it will be understood that the invention is not limited to an application to simulating the deformation of soft body tissues. As described previously, the shift in position of a rigid object (for example a bone attached to a muscle) integral with the soft tissue can also be simulated. In some cases it may be appropriate to use a torsor (of inherently known expression) rather than a deformation tensor of the type described above.

The formula for the deformation energy W , in both a triangular and a tetrahedral grid, is provided hereinbefore by way of example and depends on the material of which the object whose deformation is being simulated consists. However, it is generally expressed by a combination of the traces of the tensor E and the tensor E^2 . In these particular cases, only the trace of the tensor E (or the tensor E^2) may be used.

Claims

1. Apparatus for simulating the deformation of materials, particularly of soft body tissues, comprising:

- a memory zone (MEM, MT) adapted to store, for at least one object:

- mechanical parameters of the material of the object (λ, μ),
- data as to the position of the object (L_j), recorded at the vertices (S_j) of at least one selected mesh (T_n), and
- force data (F_j^n) which represent, in intensity and position, stress to be exerted on the object,

- and a computer (μP , MT) capable of co-operating with the memory zone (MEM) to evaluate new positions of the vertices (S_j), as a function of the stress exerted and the mechanical parameters of the material,

characterised in that the computer comprises:

- a module (12, 14, 16, 18, 20) adapted to determine repeatedly, for each mesh, the deviation between the current length (l_j) of an edge of the mesh and its previous length and/or its length at rest (L_j), and to store respective force data (F_j^n) for each vertex of the mesh, and
- a module (30, 32, 36) adapted to determine repeatedly, for each vertex, new data as to the position of this vertex ($S_j(Q_n)$) as a function of the composition of the forces exerted thereon (F_j^n), data relating to at least one previous position of the vertex ($S_j(T_n)$) and mechanical parameters of the material.

2. Apparatus according to claim 1, characterised in that for an object in the form of a hollow three-dimensional envelope the grid pattern chosen is triangular, so that the computer is able to determine the composition of forces at

each vertex of a triangle, as a function of the deviation between the current length of each side of the triangle and the length of this side at rest.

3. Apparatus according to one of claims 1 and 2, characterised in that, for an object of solid three-dimensional shape, the grid pattern chosen is tetrahedral, so that the computer is able to estimate the composition of the forces at each vertex of the tetrahedron, as a function of the deviation between the current length of each edge of the tetrahedron and the length of this edge at rest.

4. Apparatus according to one of the preceding claims, characterised in that the computer is adapted to determine differences between the squares of the current length (l_j) and the preceding length and/or the length at rest (L_j) of each edge in order to determine said composition of forces.

5. Apparatus according to one of the preceding claims, characterised in that the memory zone is adapted to store, in association with each mesh, mechanical parameters of the material of the mesh (λ_n , μ_n), at least partially defined locally, particularly at the level of the mesh or elements thereof.

6. Apparatus according to claim 5, characterised in that the computer is adapted to determine said deviation between the current and at rest lengths, with a view to estimating a derivative of the potential deformation energy (W) of each mesh, expressed as a function of a Green-St Venant tensor (E) and of mechanical coefficients inherent in the material in said mesh.

7. Apparatus according to claim 6, characterised in that the mechanical parameters comprise the Lamé coefficients of

the material in each mesh in question (λ_n , μ_n).

8. Apparatus according to one of the preceding claims, characterised in that it comprises a module for developing a data structure and adapted to delete mesh sides or edges (A_i) which connect two so-called "virtual" vertices (S_i).

9. Apparatus according to claim 8, characterised in that the module for developing the data structure is further adapted to verify that the grid pattern satisfies predefined properties of conformity.

10. Apparatus according to one of the preceding claims, characterised in that the module (30, 32, 36) adapted to determine the new positional data of the vertices ($Q'n$) as a function of the composition of forces at each vertex, is adapted to determine said new positional data as a function of time ($E3$), which makes it possible to follow the evolution of the respective positions of the vertices over time.

11. Apparatus according to claim 10, characterised in that the determination of the new positional data of the vertices ($Q'n$) uses a model for solving a differential equation applied to each vertex.

12. Apparatus according to one of the preceding claims, characterised in that the computer is capable of repeatedly determining the positional data of the vertices of the grid, with a view to determining the evolution of said positions over time ($t+\Delta t$).

13. Apparatus according to claim 12, characterised in that it comprises a display interface (VISU, IG) capable of representing the object in a predetermined form and shown

with the desired grid pattern, and in that the display interface is adapted to co-operate with the computer in order to display the shape of the moving object.

14. Apparatus according to one of the preceding claims, characterised in that it comprises a user interface (IU) provided with a handling device (CLA, MO) for simulating one or more forces exerted globally on the object.

ANNEX

F1 : $C = \nabla \Phi^t \nabla \Phi.$

5 F2 : $\Phi(X) = \sum_{j=0}^2 \Lambda_j(X) Q_j \Rightarrow \nabla \Phi = \sum_{j=0}^2 \alpha_j \otimes Q_j \text{ where } \alpha_j = \frac{l_j}{2A} n_j.$

F3 : $C = \sum_{j,k} (Q_j \cdot Q_k) (\alpha_j \otimes \alpha_k).$

10 F4 :
$$\begin{aligned} C &= \sum_j r^2 (\alpha_j \otimes \alpha_j) + \sum_{\substack{j,k \\ j \neq k}} \left(r^2 - \frac{L_p^2}{2} \right) (\alpha_j \otimes \alpha_k) \\ &= r^2 \sum_{j,k} (\alpha_j \otimes \alpha_k) - \frac{1}{2} \sum_{\substack{j,k \\ j \neq k}} L_p^2 (\alpha_j \otimes \alpha_k) \\ 15 &= -\frac{1}{2} \sum_{\substack{j,k \\ j \neq k}} L_p^2 (\alpha_j \otimes \alpha_k) \end{aligned}$$

20 F5 :
$$\begin{aligned} W &= \frac{\lambda}{2} (\text{tr } E)^2 + \mu \text{tr } E^2 \\ \text{tr } E &= \frac{1}{2} \boxed{\text{tr } C} - 1 \\ \text{tr } E^2 &= \frac{1}{4} \boxed{\text{tr } C^2} - \text{tr } E - \frac{1}{2} \end{aligned}$$

25 F6 :
$$\text{tr } C = -\frac{1}{2} \sum_{\substack{j,k \\ j \neq k}} L_p^2 (\alpha_j \cdot \alpha_k).$$

F7 :

$$\overrightarrow{P_p P_q} \cdot \overrightarrow{P_p P_r} = \begin{cases} l_j l_k \cos(\theta_p) = \frac{1}{2} (l_j^2 + l_k^2 - l_p^2) = K_p & \text{if } q \neq r, j = \text{comp}(p, q) \\ & \text{and } k = \text{comp}(p, r) \\ l_j^2 & \text{if } q = r, j = \text{comp}(p, q) \end{cases}$$

30 R1 :
$$(\alpha_j \cdot \alpha_k) = \begin{cases} -\frac{l_j l_k}{4A^2} \cos(\theta_p) = -\frac{1}{4A^2} K_p & \text{if } j \neq k \text{ and } p = \text{comp}(j, k), \\ \frac{l_j^2}{4A^2} & \text{if } j = k \end{cases}$$

F8 :

$$tr C = \frac{1}{4A^2} \sum_p L_p^2 \kappa_p.$$

F9 :

$$tr E = \frac{1}{8A^2} \sum_p (L_p^2 - l_p^2) \kappa_p = \frac{1}{8A^2} [(L_0^2 - l_0^2) \kappa_0 + (L_1^2 - l_1^2) \kappa_1 + (L_2^2 - l_2^2) \kappa_2]$$

5

F10 :

$$tr C^2 = \frac{1}{4} \sum_{\substack{j,k \\ j \neq k}} \sum_{\substack{m,n \\ m \neq n}} L_p^2 L_q^2 (\alpha_j \cdot \alpha_n) (\alpha_k \cdot \alpha_m)$$

F11 :

$$\begin{aligned} F_0^T = \frac{\partial W}{\partial Q_0} = & \frac{1}{32A^4} \left[(\lambda + \mu) \left(\sum_p (L_p^2 - l_p^2) \kappa_p \right) \left(\overrightarrow{Q_2 Q_0} \kappa_1 + \overrightarrow{Q_1 Q_0} \kappa_2 \right) \right. \\ & + \mu \left((L_1^2 - l_1^2) l_0^2 (l_2^2 \overrightarrow{Q_2 Q_0} - \kappa_0 \overrightarrow{Q_1 Q_0}) + (L_2^2 - l_2^2) l_0^2 (l_1^2 \overrightarrow{Q_1 Q_0} - \kappa_0 \overrightarrow{Q_2 Q_0}) \right. \\ & \left. \left. - (L_0^2 - l_0^2) (l_1^2 \kappa_1 \overrightarrow{Q_1 Q_0} + l_2^2 \kappa_2 \overrightarrow{Q_2 Q_0}) \right) \right] \end{aligned}$$

10 F12 :

$$\nabla \Phi = \sum_{j=0}^3 \alpha_j \otimes Q_j \quad \alpha_j = \frac{A_j}{3V} n_j$$

F13 :

$$C = \sum_{j,k} (Q_j \cdot Q_k) (\alpha_j \otimes \alpha_k)$$

15 F14 :

$$(Q_j \cdot Q_k) = \begin{cases} r^2 - \frac{L_{jk}^2}{2} & \text{if } j \neq k \\ r^2 & \text{if } j = k \end{cases} \Rightarrow \boxed{C = -\frac{1}{2} \sum_{\substack{j,k \\ j \neq k}} L_p^2 (\alpha_j \otimes \alpha_k)}$$

F15 :

$$\begin{aligned} W &= \frac{\lambda}{2} (tr E)^2 + \mu tr E^2 \\ tr E &= \frac{1}{2} \boxed{tr C} - \frac{3}{2} \\ tr E^2 &= \frac{1}{4} \boxed{tr C^2} - tr E - \frac{3}{4} \end{aligned}$$

20

F16 :

5

$$(\alpha_j \cdot \alpha_k) = \begin{cases} -\frac{1}{9V^2} \mathcal{K}_{jk} & \text{if } j \neq k \\ \frac{1}{9V^2} \mathcal{A}_j^2 & \text{if } j = k \end{cases}$$

$$\text{with } \mathcal{K}_{jk} = \frac{2 l_{jm}^2 (l_{jl}^2 + l_{kl}^2 - l_{jk}^2) - (l_{jl}^2 + l_{lm}^2 - l_{jm}^2)(l_{kl}^2 + l_{lm}^2 - l_{km}^2)}{16}$$

10

F17 :

$$\text{tr } C = \frac{1}{9V^2} \sum_{\text{Edges } (j,k)} L_{jk}^2 \mathcal{K}_{jk}$$

15 F18 :

$$\begin{aligned} \text{tr } C^2 = \frac{1}{81 V^4} & \left[\frac{1}{2} \sum_{\text{edges } (j,k)} \frac{L_{jk}^4}{2} (\mathcal{K}_{jk}^2 + \mathcal{A}_j^2 \mathcal{A}_k^2) \right. && 6 \text{ terms} \\ & + \sum_{\text{pairs of edges } ((j,k),(j,m))} \frac{L_{jk}^2 L_{jm}^2}{2} (\mathcal{K}_{jk} \mathcal{K}_{jm} - \mathcal{A}_j^2 \mathcal{K}_{km}) && 12 \text{ terms} \\ & \left. + \sum_{\text{pairs of edges } ((j,k),(m,n))} \frac{L_{jk}^2 L_{mn}^2}{2} (\mathcal{K}_{jn} \mathcal{K}_{km} + \mathcal{K}_{jm} \mathcal{K}_{kn}) \right] && 3 \text{ terms} \end{aligned}$$

F19 :

$$W = \frac{1}{324 V^4} \left\{ \left(\frac{\lambda + \mu}{2} \right) \left[\sum_{\text{Edges } (j,k)} (L_{jk}^2 - l_{jk}^2) \mathcal{K}_{jk} \right]^2 + \mu \tilde{\Delta} \right\}$$

20

F20 :

$$\begin{aligned} \tilde{\Delta} = & \frac{1}{2} \sum_{\text{edges } (j,k)} (L_{jk}^2 - l_{jk}^2)^2 \mathcal{A}_j^2 \mathcal{A}_k^2 \\ & - \sum_{\text{pairs of edges } ((j,k),(j,m))} (L_{jk}^2 - l_{jk}^2)(L_{jm}^2 - l_{jm}^2) \mathcal{A}_j^2 \mathcal{K}_{km} \\ & + \sum_{\text{pairs of edges } ((j,k),(m,n))} (L_{jk}^2 - l_{jk}^2)(L_{mn}^2 - l_{mn}^2) (\mathcal{K}_{jn} \mathcal{K}_{km} + \mathcal{K}_{jm} \mathcal{K}_{kn} - \mathcal{K}_{jk} \mathcal{K}_{mn}) \end{aligned}$$

30

F21 :

$$\begin{aligned}
F_p = \frac{1}{162 V^4} & \left\{ (\lambda + \mu) \left[\sum_{\text{Edges}(j,k)} (L_{jk}^2 - l_{jk}^2) \kappa_{jk} \right] \left(\sum_{\text{Edges}(j,p)} \overrightarrow{Q_j Q_p} \kappa_{jp} \right) \right. \\
& + \mu \left[\sum_{\text{Edges}(j,p)} (L_{jp}^2 - l_{jp}^2) A_p^2 \left(\overrightarrow{Q_j Q_p} A_j^2 - \overrightarrow{Q_k Q_p} \kappa_{jk} - \overrightarrow{Q_m Q_p} \kappa_{jm} \right) \right. \\
& \left. \left. + \sum_{\substack{\text{Edges}(j,k) \\ j,k \neq p}} (L_{jk}^2 - l_{jk}^2) \left(\overrightarrow{Q_m Q_p} \kappa_{jkmp} - \overrightarrow{Q_j Q_p} A_j^2 \kappa_{kp} - \overrightarrow{Q_k Q_p} A_k^2 \kappa_{jp} \right) \right] \right\}
\end{aligned}$$

10

$$\begin{aligned}
R2 : \quad W &= \frac{1}{128 A^4} \left[(\lambda + \mu) \left(\sum_p (L_p^2 - l_p^2) \kappa_p \right)^2 \right. \\
& \left. + \mu \left(\sum_p (L_p^2 - l_p^2)^2 l_j^2 l_k^2 - 2 \sum_{\substack{j,k \\ j \neq k}} (L_j^2 - l_j^2) (L_k^2 - l_k^2) l_p^2 \kappa_p \right) \right] \\
& p = \text{comp}(j, k)
\end{aligned}$$

15

$$E1 : \quad m_i \frac{d^2 P_i}{dt^2} = \gamma_i \frac{dP_i}{dt} + F_i$$

20

$$E2 : \quad M\ddot{U} + C\dot{U} + KU = R$$

25

$$E3 : \quad \left(\frac{m_i}{\Delta t^2} - \frac{\gamma_i}{2\Delta t} \right) P_i^{t+1} = F_i + \frac{2m_i}{\Delta t^2} P_i^t - \left(\frac{m_i}{\Delta t^2} + \frac{\gamma_i}{2\Delta t} \right) P_i^{t-1}$$